

Fig. 1

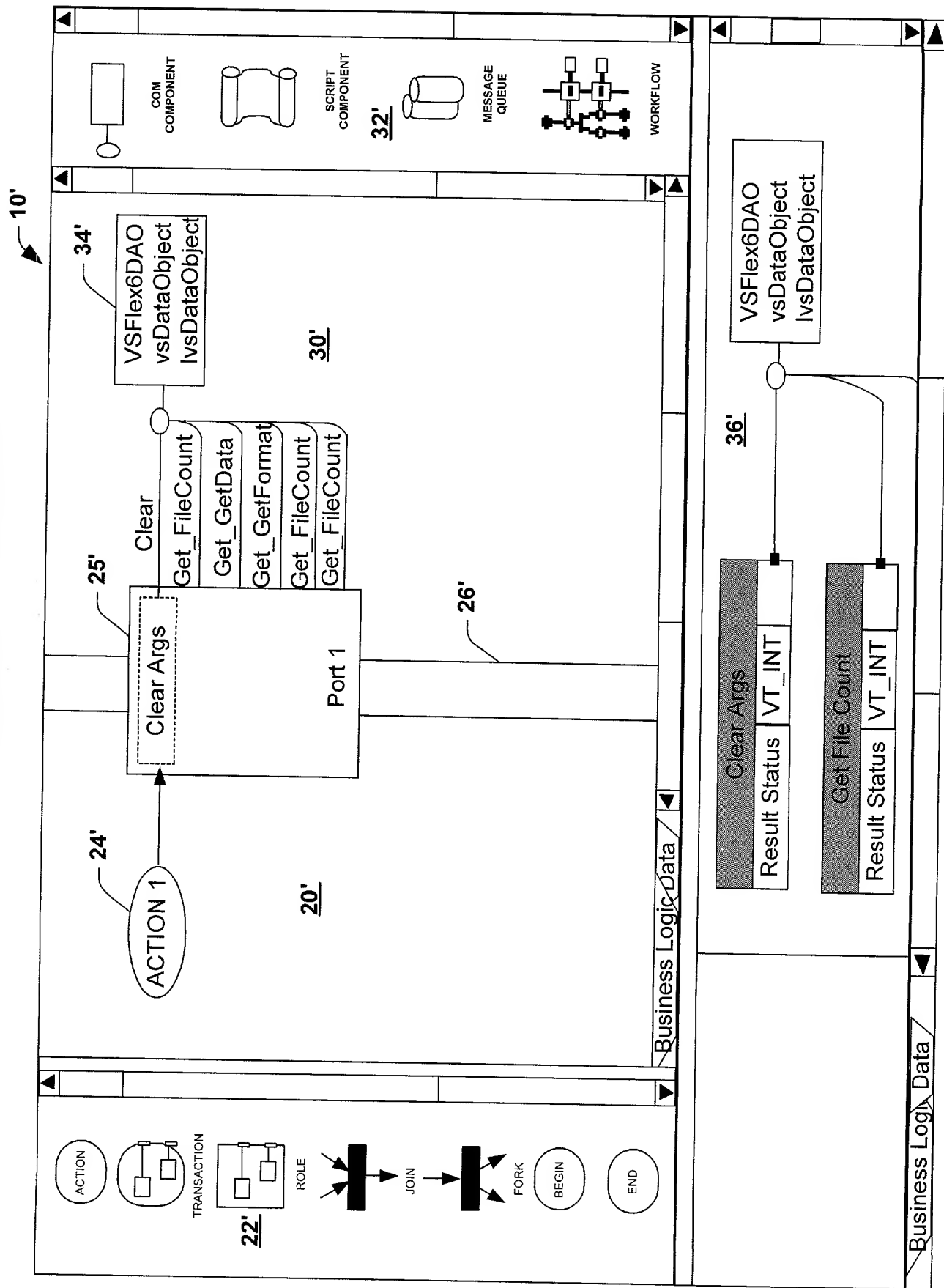


Fig. 2

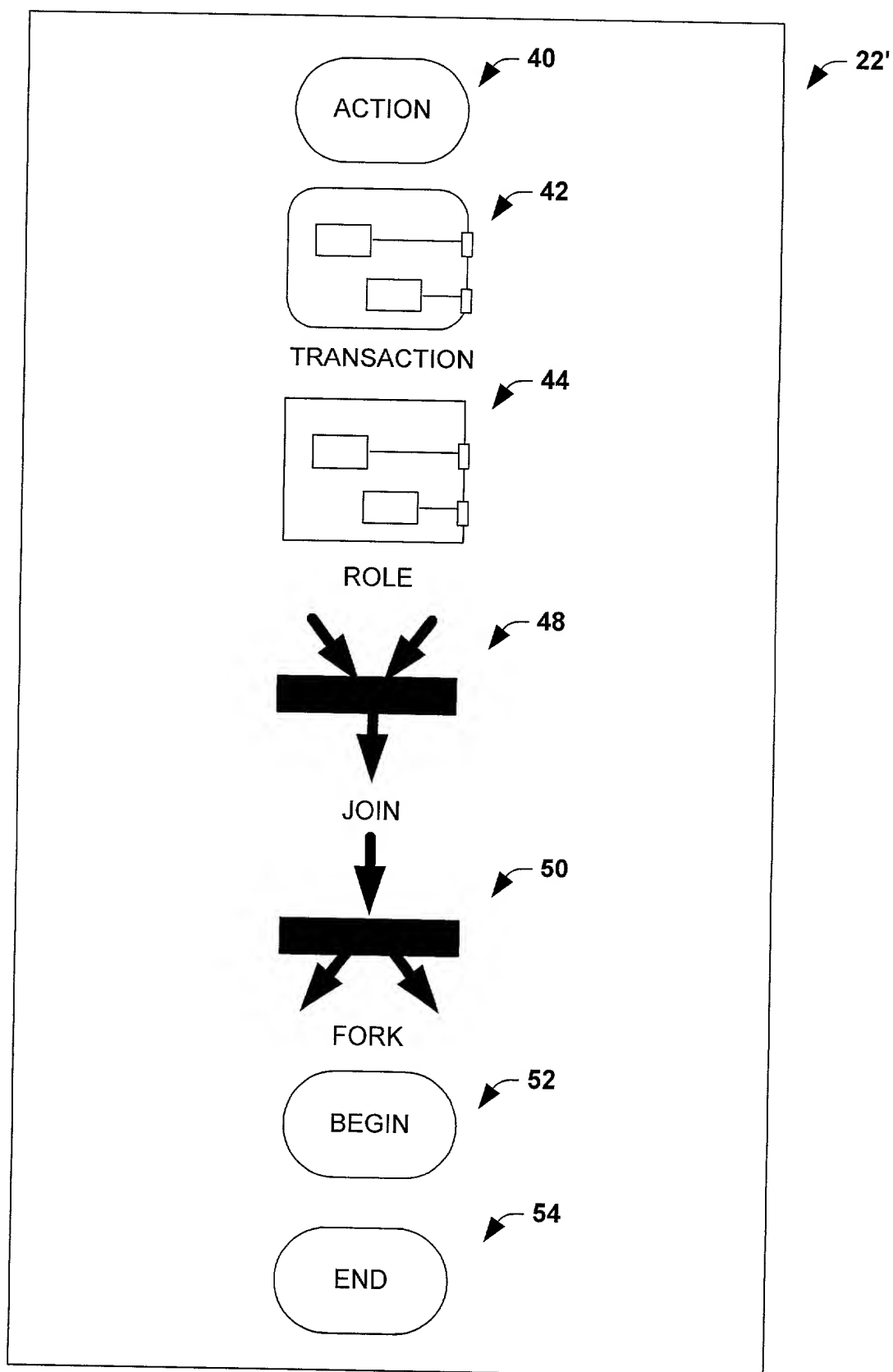
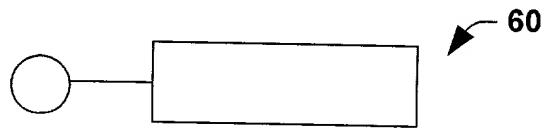
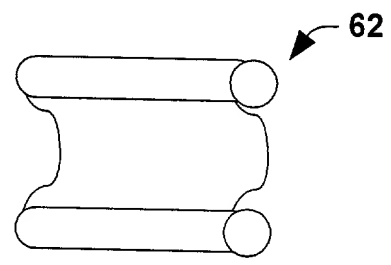


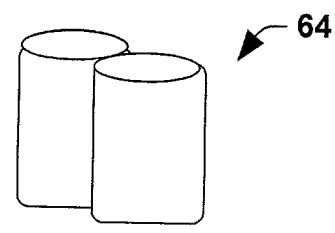
Fig. 3



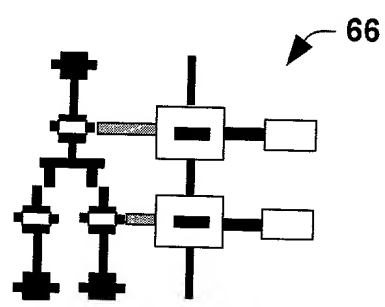
COM COMPONENT



SCRIPT COMPONENT



MESSAGE QUEUE



WORKFLOW

Fig. 4

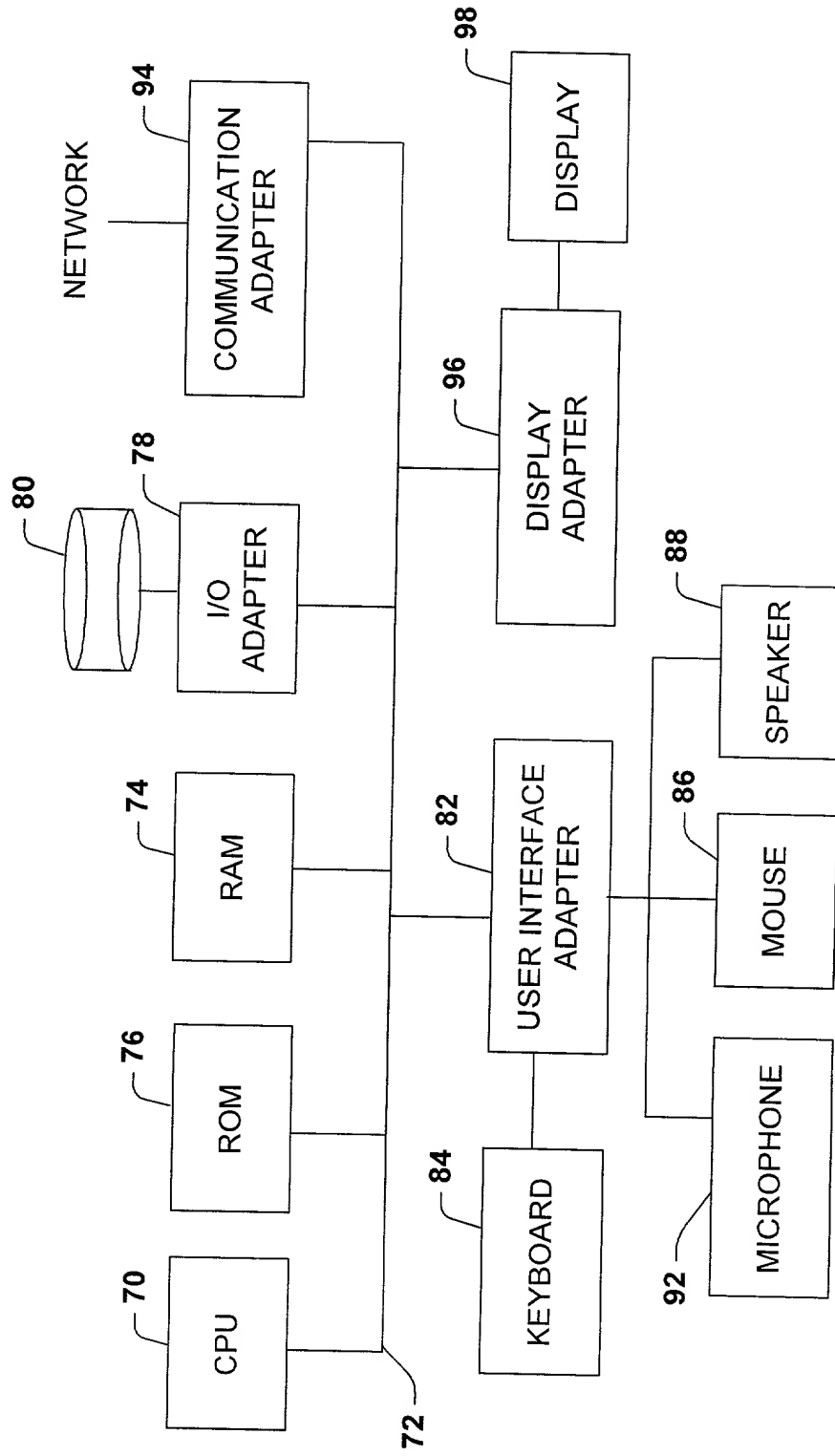


Fig. 5a

FIG. 5b

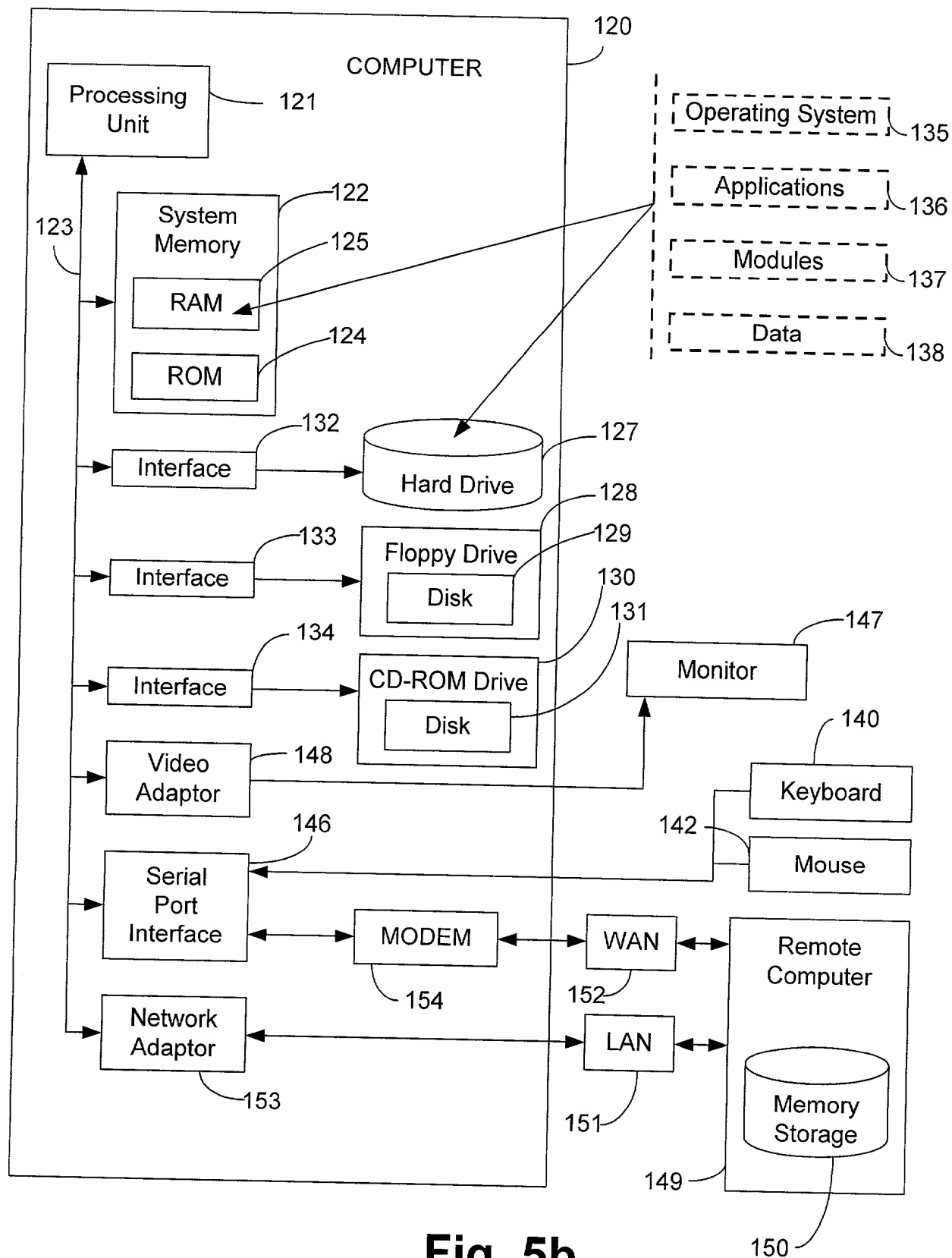


Fig. 5b

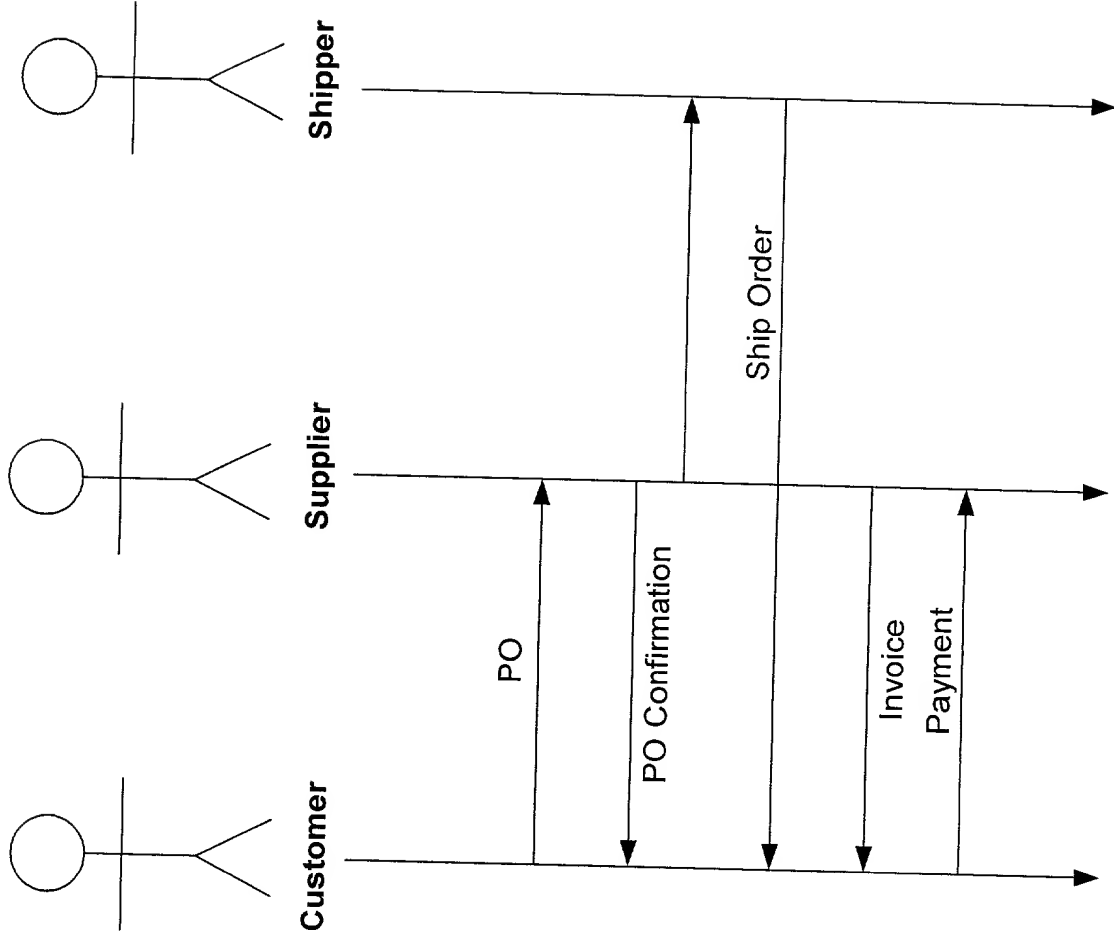


Fig. 6a

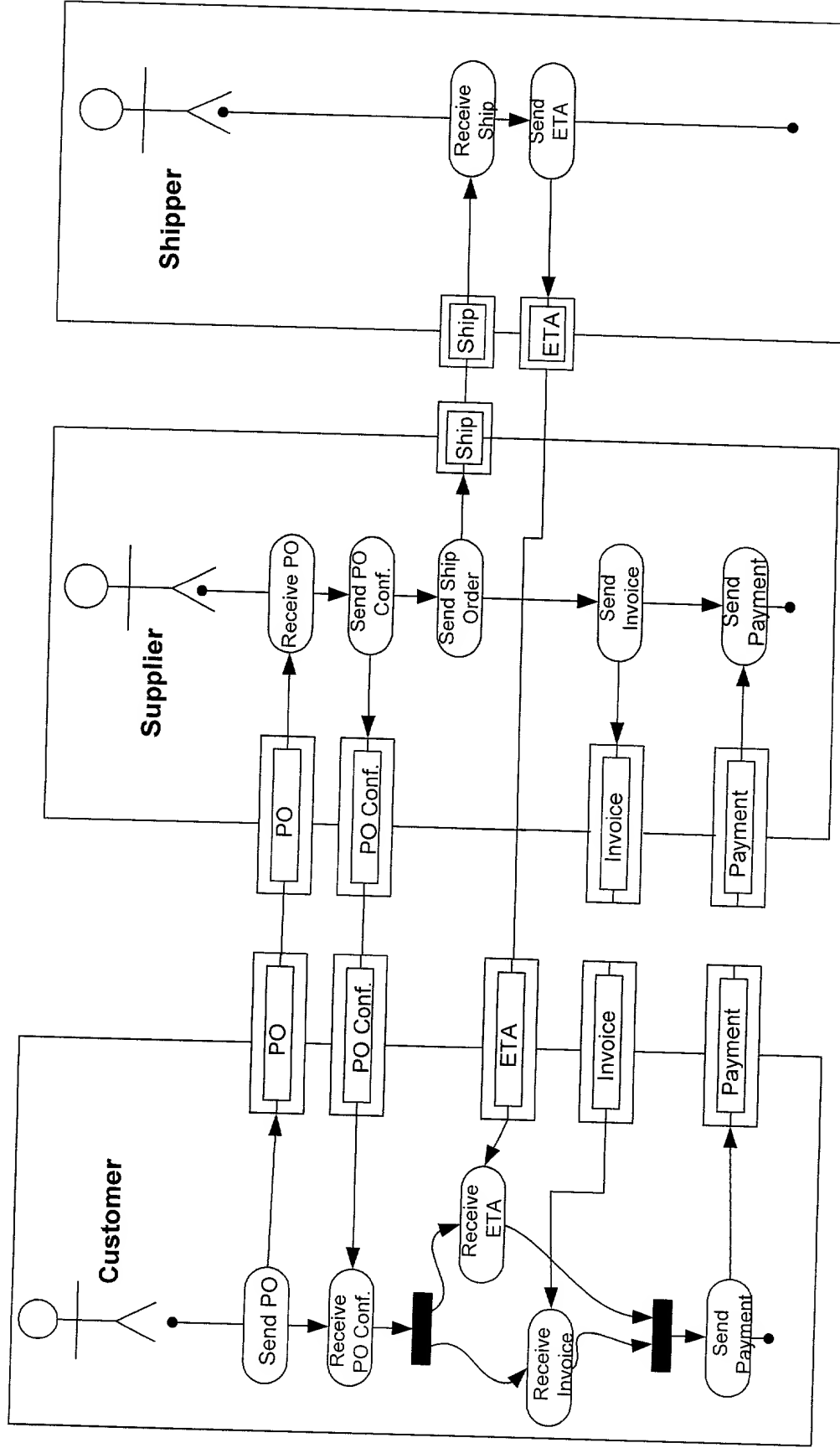


Fig. 6b

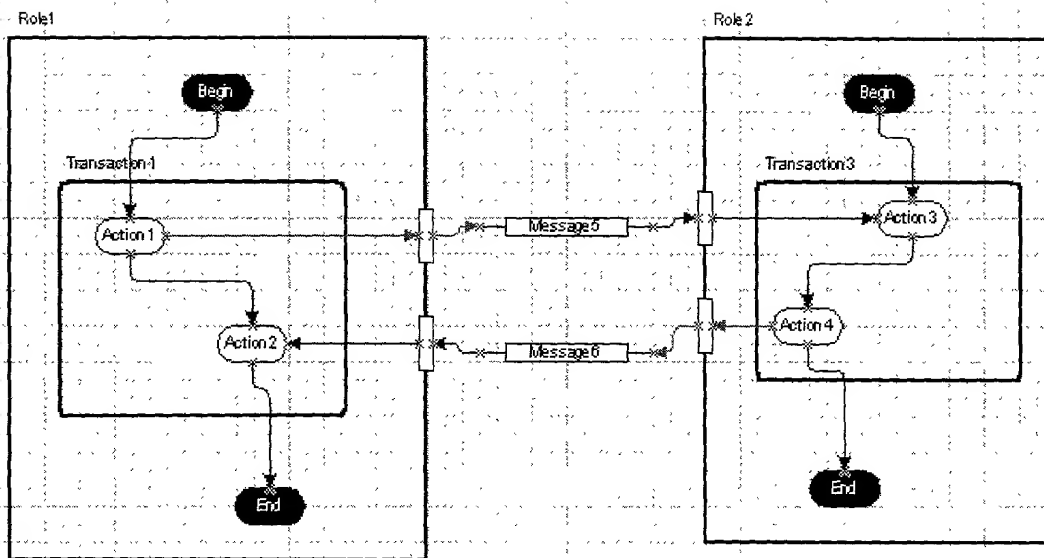


Fig. 7a

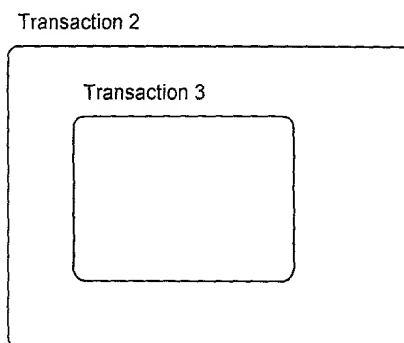


Fig. 7b

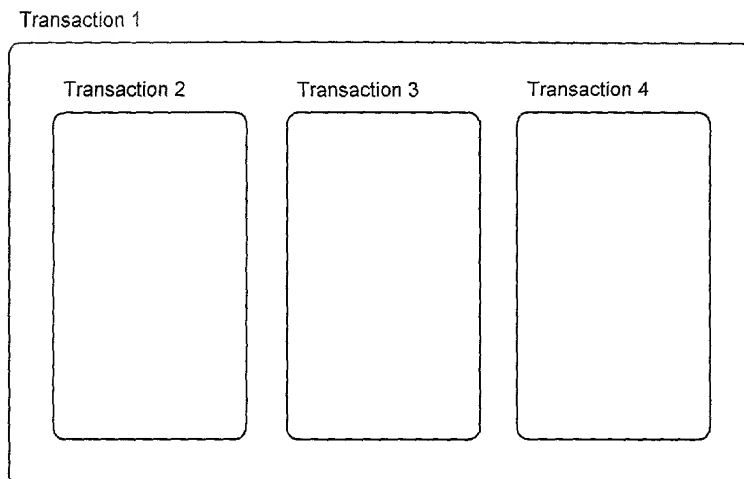


Fig. 7c

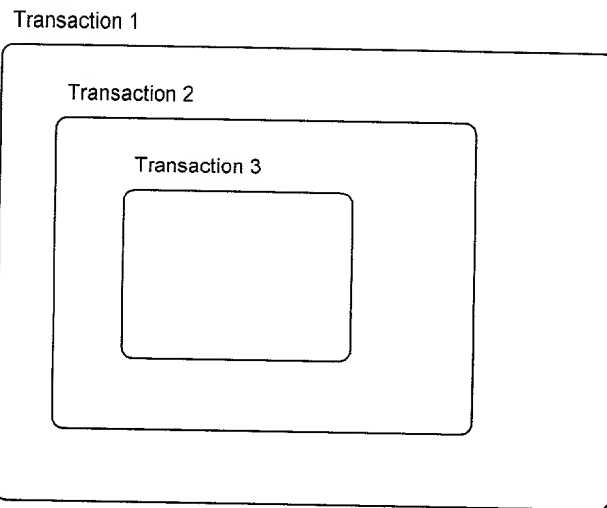


Fig. 7d

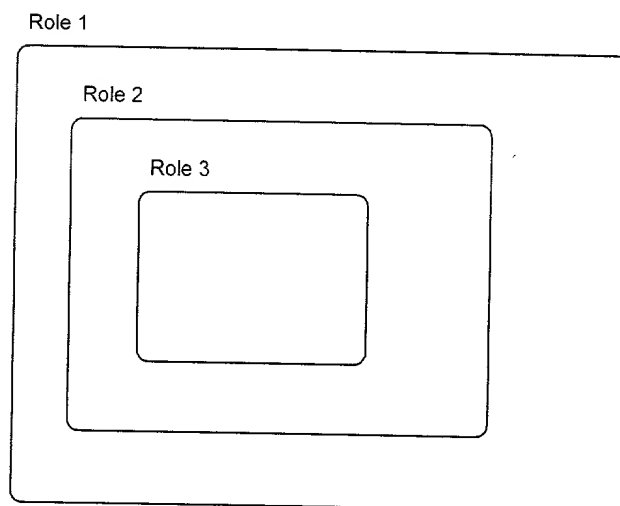


Fig. 7e

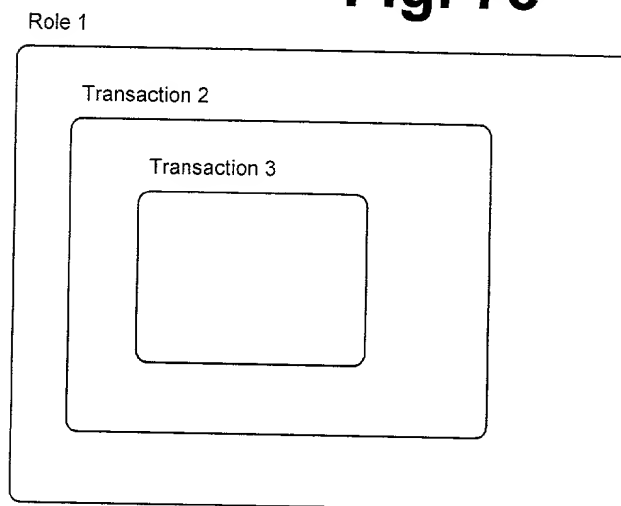


Fig. 7f

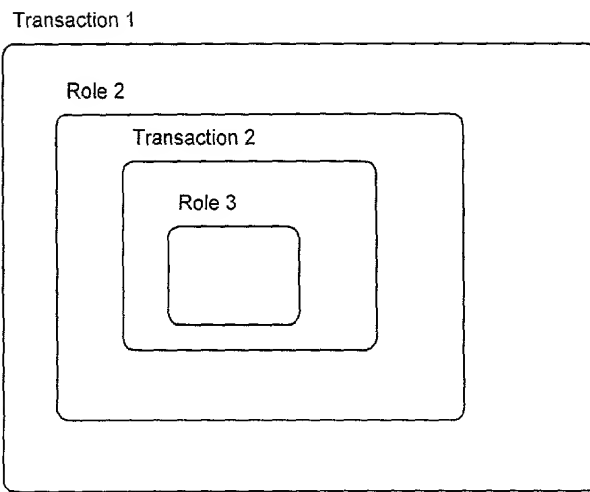


Fig. 7g

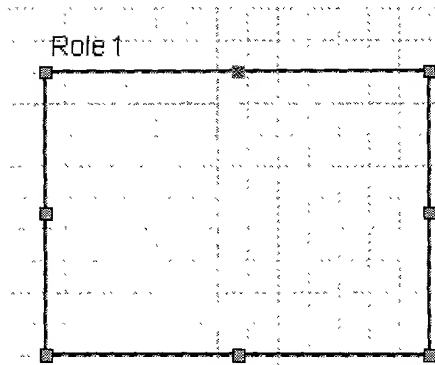


Fig. 8a

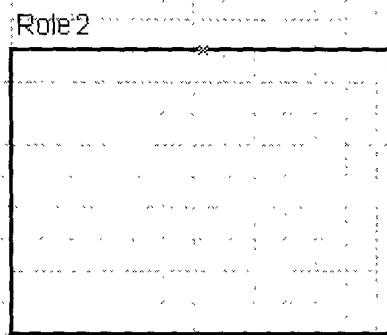


Fig. 8b

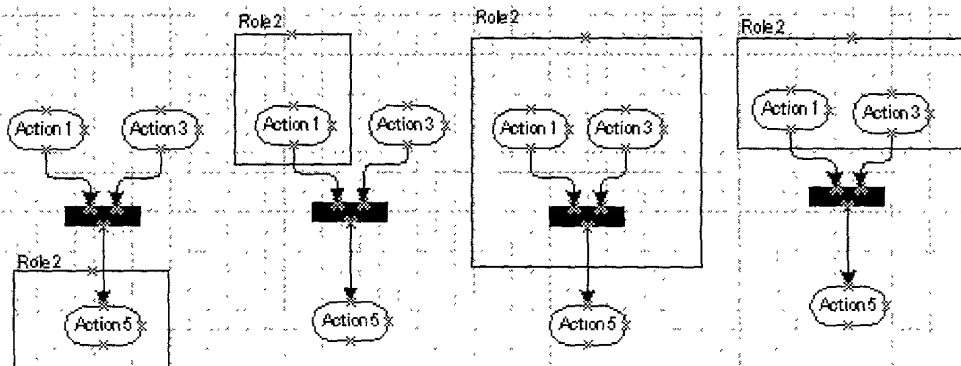


Fig. 8c

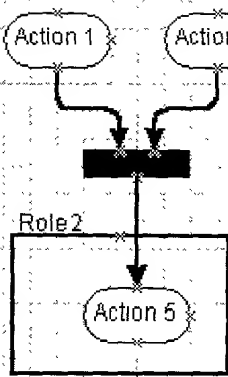


Fig. 8d

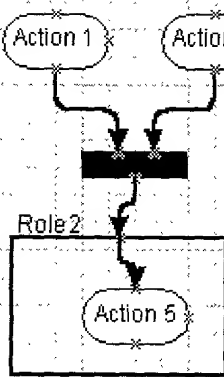


Fig. 8e

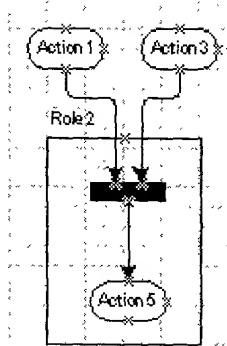


Fig. 8f

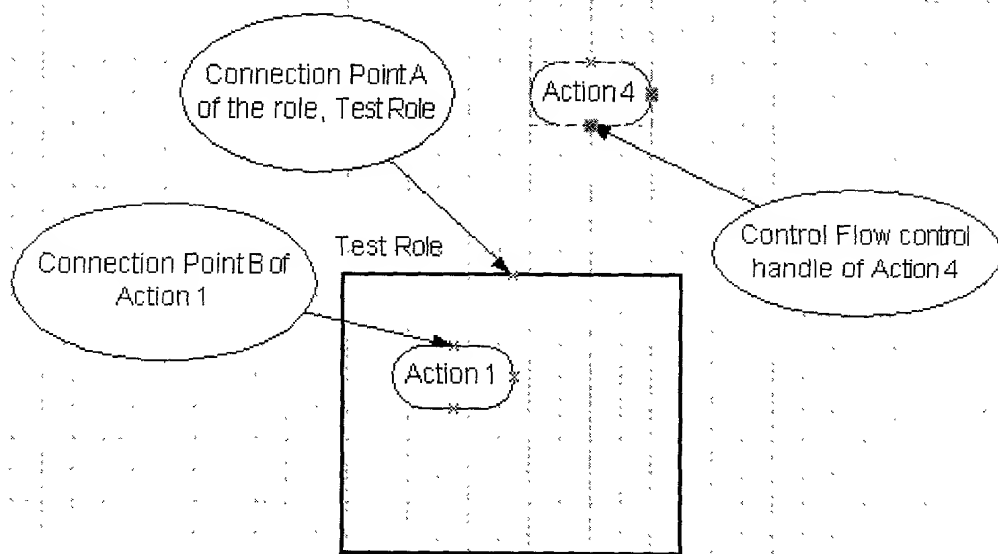


Fig. 9

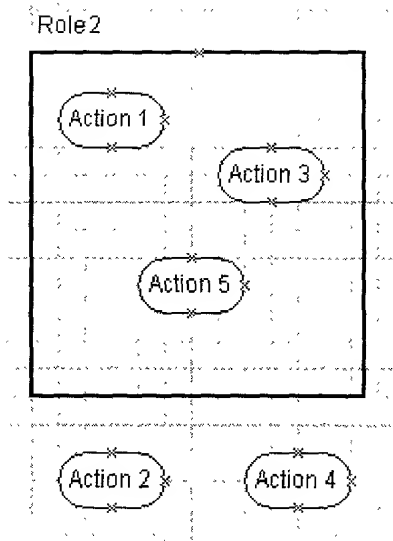


Fig. 10

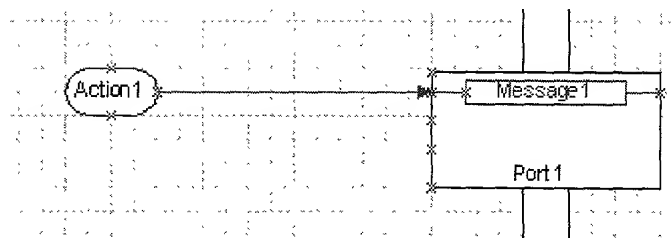


Fig. 11a

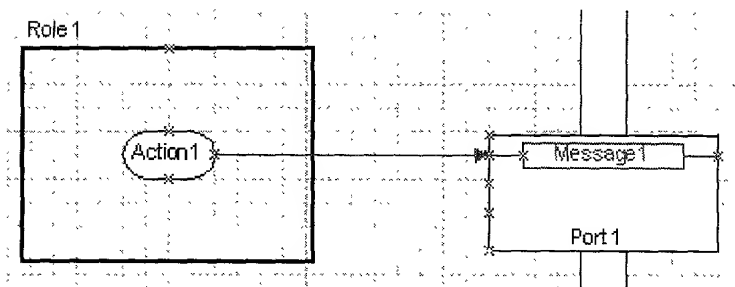


Fig. 11b

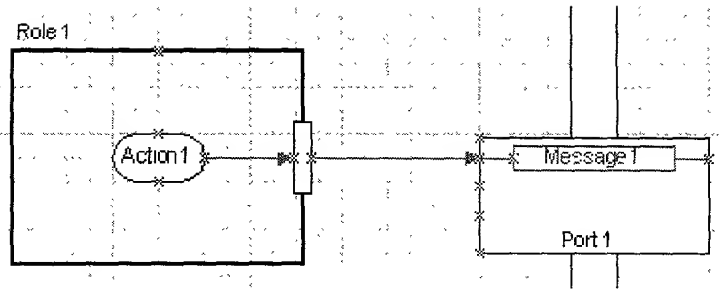


Fig. 11c

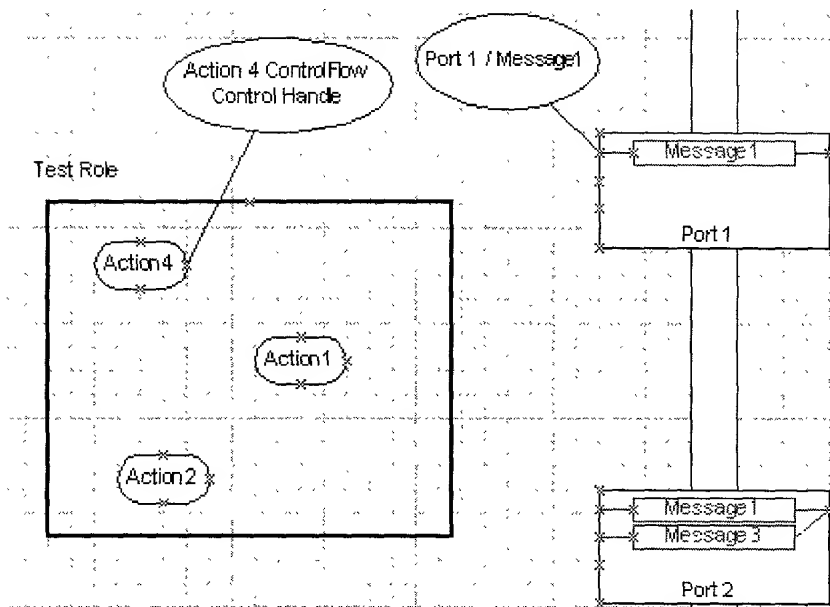


Fig. 12a

Downloaded from www.ascelibrary.org

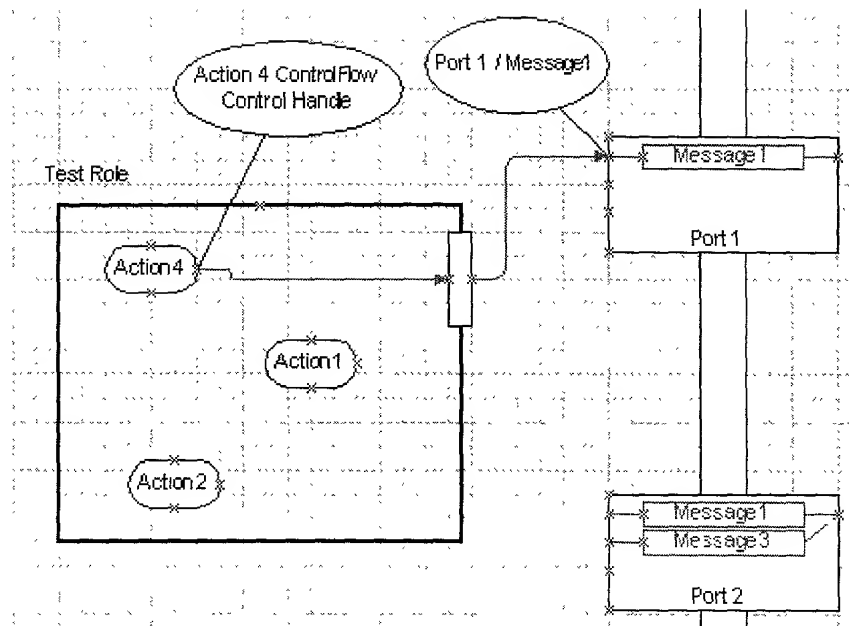


Fig. 12b

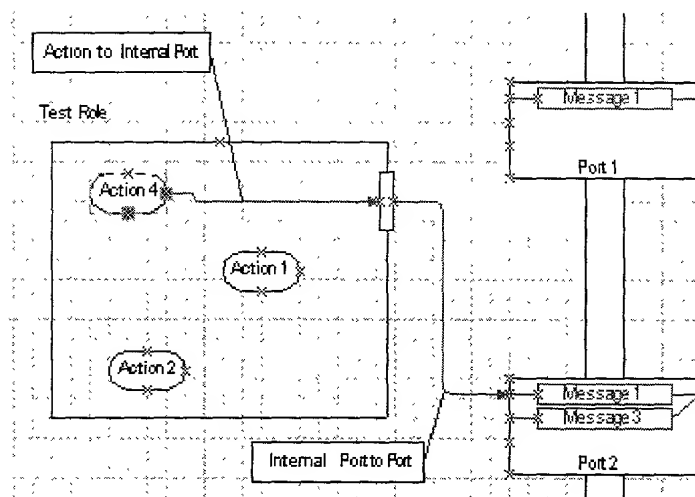


Fig. 13

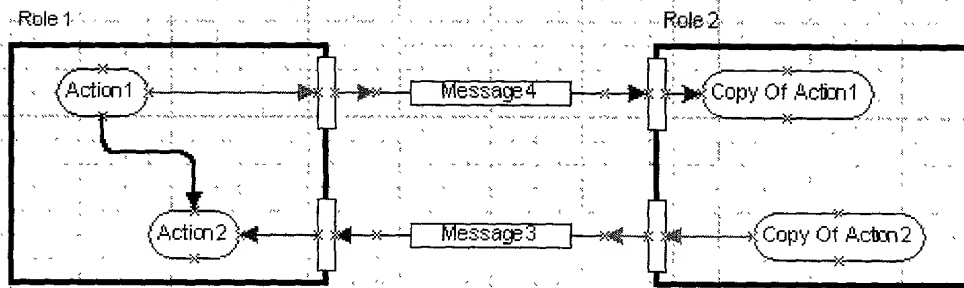


Fig. 14a

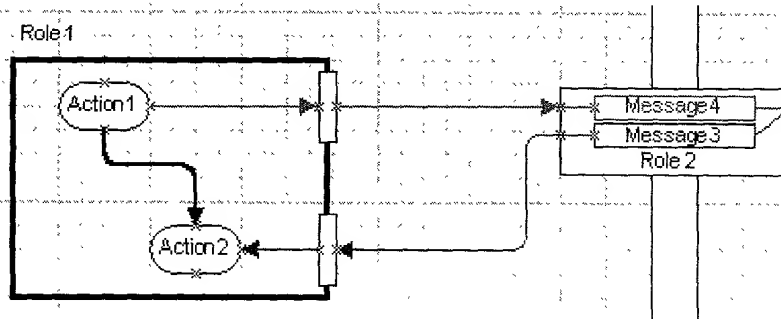


Fig. 14b

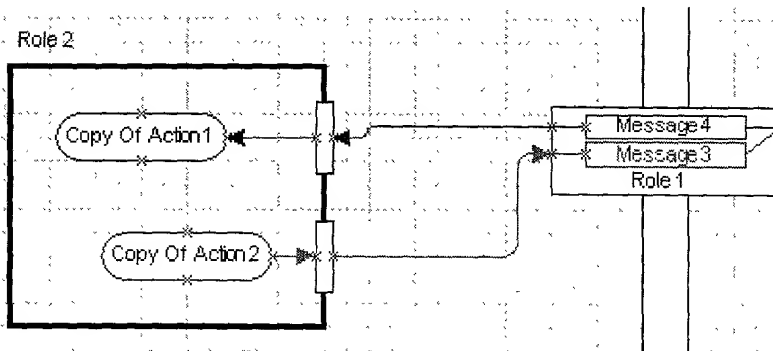


Fig. 14c

Transaction Properties

Property	Value
Name	Transaction 1
Create new instances as needed	False

Catch
☒ Enabled
Add Code

Compensation
☐ Enabled
Delete Code

OK
Cancel
Help

Fig. 15

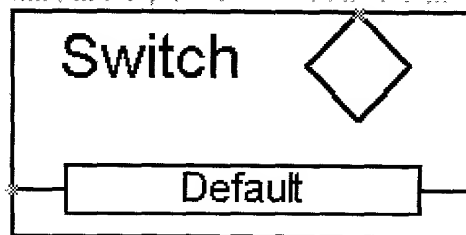


Fig. 16a

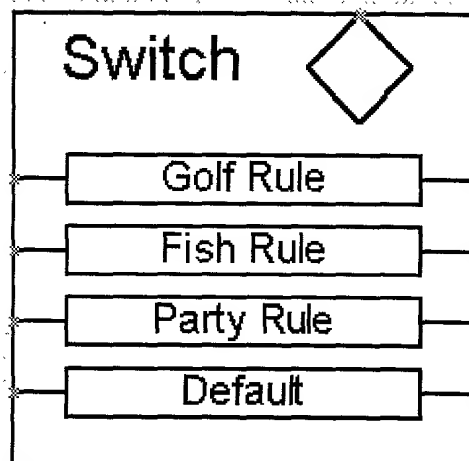


Fig. 16b

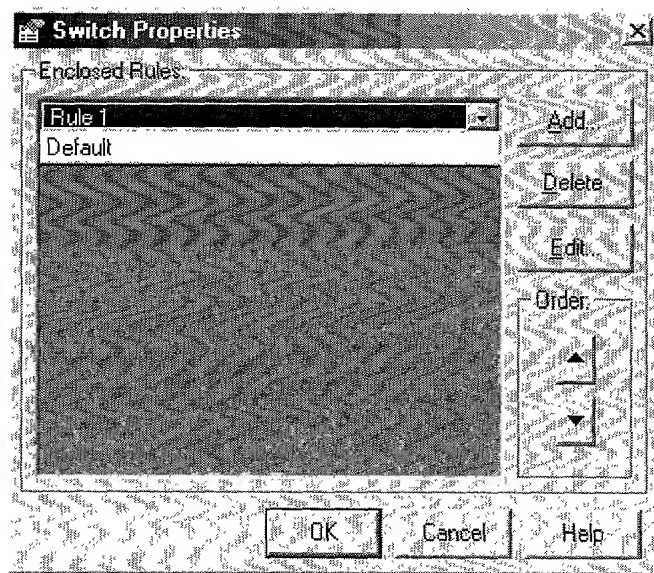


Fig. 17

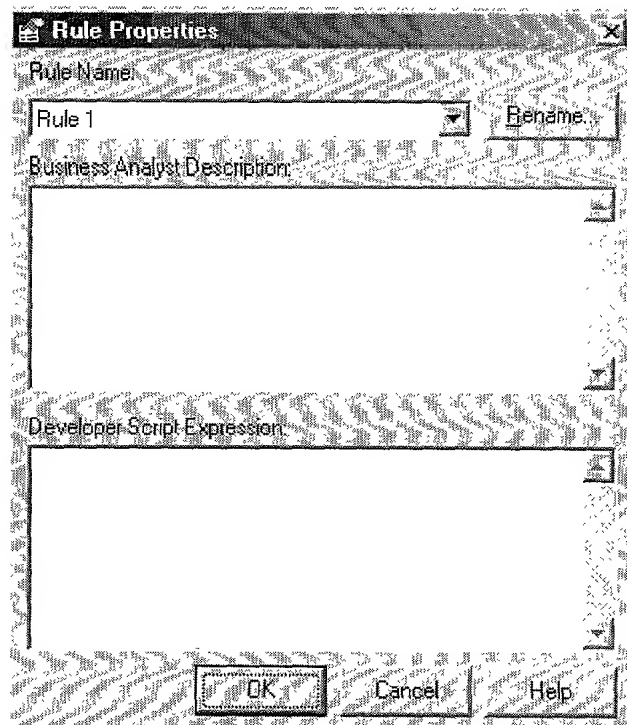


Fig. 18

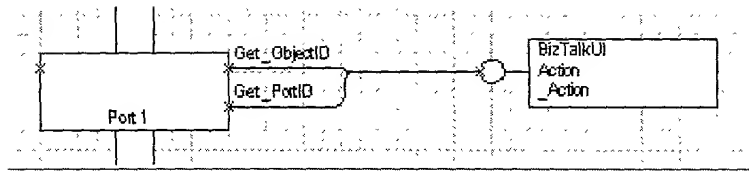


Fig. 19a

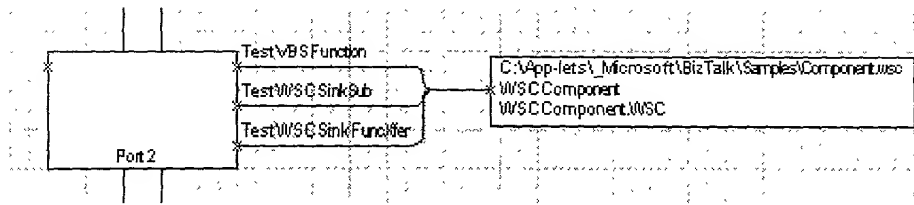


Fig. 19b

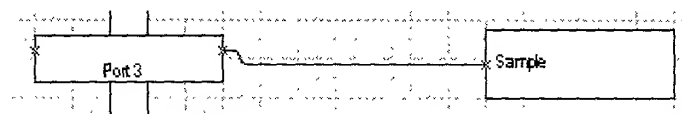


Fig. 19c

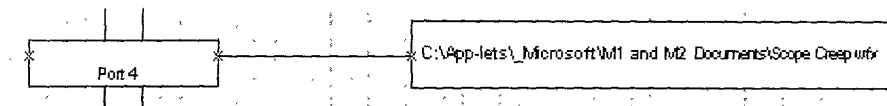


Fig. 19d

Message Properties

Message Name:

Select Method:

Field Properties Derived from Method Arguments:

Name	Data Type	Default Value
Result Status	VT_INT	
Index	VT_I2	0

Fig. 20a

Message Properties

Message Name:

XML Schema:

XML Structure:

- ☒ XML
 - ☒ ThisDOMNode1
 - ☒ Parameter #1
 - ☐ Parameter #2
 - ☐ ThisDOMNode2
 - ☐ Parameter #1
 - ☒ Parameter #2

Name	Data Type	Default Value
Result Status	VT_INT	
XML ThisDOMNode1	IDom	
XML ThisDOMNode1 Parameter #1	VT_INT	5
XML ThisDOMNode2 Parameter #2	VT_BSTR	DefaultString

Order:

↑

↓

Fig. 20b

00000153-030604

Message Properties

Message Name:
CallMessageExample Args

CALL Parameters:

Ports:

Called Schedule	Current Schedule
Microsoft	Port 1
Dell	

Messages:

Called Schedule	Current Schedule
Send PO	Message 1
Receive Acknowl	Message 2

Transactions:

Called Schedule	Current Schedule
Transaction 1	Transaction 2
Transaction 2	Transaction 1
Transaction 3	Transaction 3
Transaction 4	

OK
Cancel
Help

Fig. 20c

Port References Message Properties

Enclosed Ports:

Port 1
Port 2
Port 3
Port 4

Edit...

Order:

OK
Cancel
Help

Fig. 20d

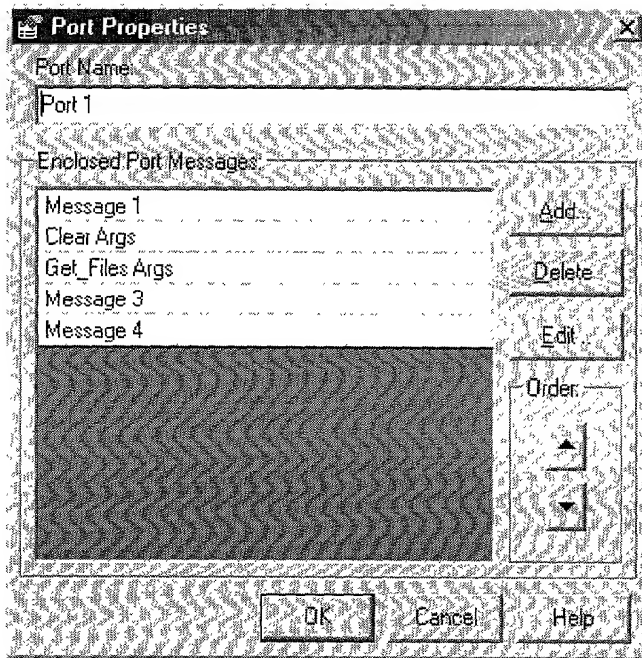


Fig. 21

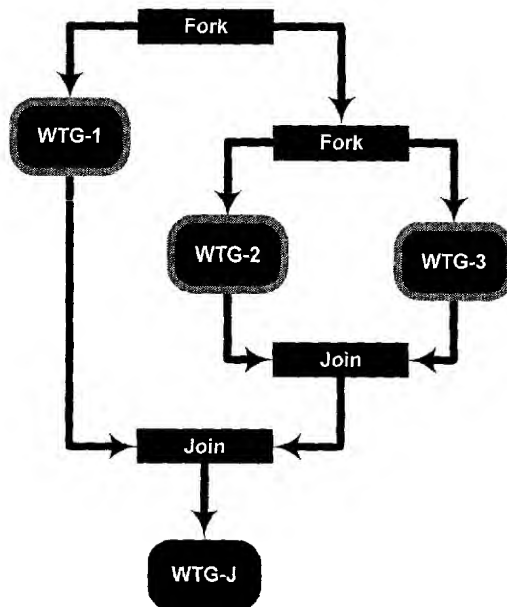


Fig. 22

```

<connect>
  <partition>
     $\sigma(\delta(\text{WTG}_1, \text{source}(p_1, m_1)))$ 
    <connect>
      <partition>
         $\sigma(\delta(\text{WTG}_2, \text{source}(p_2, m_2)))$ 
         $\sigma(\delta(\text{WTG}_3, \text{source}(p_3, m_3)))$ 
      </partition>
      <sequence>
        <task choice="all">
           $\sigma(\text{sink}(p_2', m_2'))$ 
           $\sigma(\text{sink}(p_3', m_3'))$ 
        </task>
        <zero/> <!-- This is the inner Continuation -->
      </sequence>
      <connectionList>
        ports  $p_2$  and  $p_3$  with ports  $p_2'$  and  $p_3'$ 
      </connectionList>
    </connect>
  </partition>
  <sequence>
    <task choice="all">
       $\sigma(\text{sink}(p_1', m_1'))$ 
    </task>
     $\sigma(\text{WTG}_j)$  <!-- Outer Continuation -->
  </sequence>
  <connectionList>
    port  $p_1$  with port  $p_1'$ 
  </connectionList>
</connect>

```

Fig. 23

SLANG

schedule	::= header? process? contextRef?
header	::= portList? messageList? contextList?
process	::= zero sequence switch map copy partition connect cut
portList	::= port*
messageList	::= message*
contextList	::= context*
zero	::= zero
sequence	::= genericAction* process? contextRef?
genericAction	::= silence action task call return release
silence	::= silence
action	::= source sink
source	::= portRef messageRef contextRef?
sink	::= portRef messageRef contextRef?
task	::= action* contextRef?
call	::= schedRef portRef* messageRef* contextRef?
switch	::= branch* (default process) ? contextRef?
branch	::= case process
case	::= ruleRef msgRef msgRef
map	::= process assignmentList? contextRef?
assignmentList	::= assignment*
assignment	::= messageRef portRef
copy	::= process contextRef?
partition	::= process* contextRef?
connect	::= process process connectionList contextRef?
connectionList	::= connection*
connection	::= portRef portRef
cut	::= process process process contextRef?

Fig. 24